



Instituto Tecnológico de Costa Rica

Carrera de Computación, San Carlos

Taller de Programación

*Proyecto de Reconocimiento de las emociones*

Profesor:

MTBD. Efrén Jiménez Delgado

Integrantes del equipo:

Ashley Samantha Acuña Montero

Katherine Dayana Amador González

Campus Tecnológico Local San Carlos

## Resumen

- Motivación y problema: Lograr reforzar los conocimientos adquiridos en clase, además de implementarlos y poder crear una base de datos con estos. Crear una base de datos que permita archivar diferentes tipos de información de pacientes y médicos, que serían usuarios de la plataforma.
- El objetivo de este proyecto es crear un algoritmo que pueda ayudar a un hospital privado o público a reconocer y almacenar la información de cada uno de los usuarios registrados en la base de datos mediante una imagen o sus datos.
- Enfoque propuesto para resolver el problema: Crear un sistema donde el usuario puede registrarse e iniciar sesión dependiendo si es un médico o un paciente, este debe llevar un historial de las citas realizadas, además de poder analizar al paciente por medio de su rasgo facial para ver que enfermedades tiene, también, se puede realizar una facturación dependiendo del tiempo que dedico el médico al paciente, finalmente, realizar una estadística que puede ser accedida solo por el médico y así conocer el total de pacientes atendidos, enfermedades atendidas en total, enfermedades atendidas por género y un diagnóstico de las emociones o sentimientos mas presentes en el paciente, por último una comparación de pacientes sin problemas y pacientes con enfermedades.
- ¿Qué hizo para probar / validar su propuesta? Se utilizó Python, para la creación del sistema, por medio de algoritmos, que hicieran posible su funcionamiento, verificando que es mejor para el usuario, de manera que sea mas agradable la experiencia de la consulta, además para validar la propuesta, se realizaron multiples intentos con diferentes personas y datos.
- Los resultados de su prueba / validación: Se obtuvieron los resultados esperados en un 99 %, esto se logro por medio de constantes intentos desde el principio del algoritmo hasta al final, realizando cambios para que el algoritmo quedara mas agradable a la vista, sin necesidad de ser tan extenso a la hora de ser visto por alguien más.

# Índice general

<b>1. Introducción</b>	<b>1</b>
<b>2. I Etapa</b>	<b>3</b>
2.1. Identificación y análisis del problema . . . . .	3
2.2. Solución del problema . . . . .	4
2.3. Análisis de resultados . . . . .	4
2.4. Discusión . . . . .	1
<b>3. II Etapa</b>	<b>2</b>
3.1. Identificación y análisis del problema . . . . .	2
3.2. Solución del problema . . . . .	3
3.3. Análisis de resultados . . . . .	4
3.4. Discusión . . . . .	4
<b>4. III Etapa</b>	<b>2</b>
4.1. Identificación y análisis del problema . . . . .	2
4.2. Solución del problema . . . . .	3
4.3. Análisis de resultados . . . . .	3
4.4. Discusión . . . . .	1
<b>5. Conclusiones</b>	<b>3</b>
<b>Referencias</b>	<b>4</b>

# Índice de figuras

2.1. Diagrama de flujo de la aplicación (I Etapa). . . . .	5
3.1. Diagrama de flujo de la aplicación (II Etapa). . . . .	3
4.1. Diagrama de flujo de la aplicación (III Etapa). . . . .	3

# Índice de cuadros

2.1. Análisis de resultados con respecto a los requerimientos (I Etapa).	1
3.1. Análisis de resultados con respecto a los requerimientos (II Etapa).	4
4.1. Análisis de resultados con respecto a los requerimientos (III Etapa).	1

# Capítulo 1

## Introducción

El reconocimiento facial se ha ido implementado con el paso del tiempo, ya que, es una herramienta muy útil, debido a que permite la evaluación de los rasgos de los usuarios de alguna entidad o empresa, por medio de la comparación de imágenes evaluando los rostros de la base de datos de rostros que poseen para identificar quien es la persona y así darle acceso a información y empresas corporativas, dándose así un sistema de seguridad biométrica, ofreciendo muchas ventajas como el poder una opinión en una corte de justicia, para poder identificar si alguien es inocente o culpable, además, como lo menciona, Moreano, (2017) "La biometría está diseñada para que tú no tengas que hacer nada. Simplemente el dispositivo te reconoce"(p.144).

En la ciudad de Panamá, Panamá, el aeropuerto internacional de Tocumen consta con un sistema que realiza el reconocimiento facial de los pasajeros que ingresan a la terminal a fin de contrastarlos con las bases de datos policiales internacionales, además, Google, quien ha patentado un sistema que le permite al usuario desbloquear los dispositivos utilizando reconocimiento facial. (Poveda y Merchán, 2015).

*El objetivo de este proyecto es la creación de un código donde un usuario pueda registrarse e iniciar sesión en base a sus preferencias, llevando así el historial de el médico y el paciente, en el historial del paciente se guardaran todas las citas que ha realizado, así como el análisis realizado a estos por medio de una imagen de su rostro, este sistema permitirá analizar sus emociones y sentimientos, dando así un resultado de su enfermedad, el mismo sistema le dará al paciente el cobro de su consulta, dependiendo del tiempo que le dedico el médico al paciente y al finalmente podrá generar estadísticas de pacientes atendidos, de enfermedades en general o por género, estas*

*estadísticas solo son accesibles por los médicos.*

Para este proyecto se presentan, 3 etapas a realizar, en la primera se debe crear un sistema de seguridad, donde se registren dos tipos de usuario, ya sea médico con las características: nombre, apellido, código de médico, código de rostro, contraseña y salario por hora, o como paciente con las características: nombre, apellido, dirección, género, fecha de nacimiento, código de rostro, número de cédula, ya una vez realizado el registro pasamos al inicio de sesión, este lo pueden realizar de dos formas, por medio de usuario y contraseña o una una imagen si fuera médico y con el número de cédula o una imagen si fuera un paciente.

En la segunda etapa se debe de realizar un historial de paciente médico, donde indique todas las citas médicas que ha realizado un paciente con los datos: fecha, paciente, médico, parte médico, emociones y características de la imagen, un historial de atención médico donde pueda saber todas las características de un paciente que ha hecho una consulta anteriormente con los datos: fecha: paciente, médico, parte médico, emociones, características de la imagen, cantidad de minutos de la atención, precio por hora y precio total, y una de parte médico dando así un diagnóstico de la enfermedad variando de las emociones o sentimientos presentados en el historial de médico paciente, por ejemplo: si la emoción "anger" es arriba de 50 % y "disgust." arriba de 25 %, la enfermedad es problema de tiroides.

Finalmente, se realiza una facturación de la consulta realizada pidiendo los datos de: cobro de la consulta, y se realiza un cobro del 13 % por impuestos y por último una estadística que solo sea accesible para los médicos donde podrán observar: total de pacientes atendidos, enfermedades atendidas, enfermedades por género, sentimientos más presentes en los diagnósticos y una comparación de pacientes sin problemas y pacientes con enfermedades.

Se realizó un algoritmo en Python, para construir el sistema, en este se agregaron los conocimientos adquiridos en clase, sobre listas, diccionarios, variables, funciones, archivos, ordenamientos, además, para comprobar que no habían errores en el algoritmo se realizaron muchas pruebas, por medio de debug y corridas del código, para verificar si estaba correcto ó encontrar errores en este.

Este documento está organizado de la siguiente forma: en el capítulo 2 se tiene que realizar un código de seguridad que consiste en registrar e iniciar sesión de un usuario, donde este tiene que tener ciertas características dependiendo si es médico o paciente, en el capítulo 3 se describe la propuesta y solución a la resolución de la segunda etapa, que se basa en almacenar los

datos del médico y paciente en un historial, además de una parte médico, donde se realiza un análisis del rostro del paciente por medio de una imagen, dando así el resultado de su posible enfermedad si fuera el caso, en el capítulo [4](#) se describe la tercera etapa del proyecto se realiza una facturación variando de la consulta y el tiempo que duro el paciente, de este modo, facturando el precio mas un 13 % de impuestos, por ultimo se crean estadísticas donde el medico puede acceder, dejando analizar las consultas realizadas en el hospital por parte de un paciente y sus diagnosticos, en el capítulo [5](#) se presenta lo aprendido al final de la realización del proyecto por medio de conclusiones.



# Capítulo 2

## I Etapa

### 2.1. Identificación y análisis del problema

Implementar algoritmos en Python, que consisten en dos partes: la seguridad donde tenga la posibilidad de registrarse e iniciar sesión como un paciente o como médico siendo estos usuarios: un médico con las características de: nombre, apellido, código de médico, código de rostro, contraseña y salario por hora, o un paciente con las características de: nombre, apellido, dirección, género, fecha de nacimiento, código de rostro, dando así una herramienta necesaria para manejar mejor la organización de los usuarios, del mismo modo que este sea mas sencillo y organizado sin necesidad incurrir en el gasto del tiempo a una entidad pública o privada.

Para que este funcione, se necesitara crear un menú, que tenga diferentes opciones para el usuario, desde el registrarse hasta el inicio de sesión, si este no esta registrado, los datos no serán brindados y no podrá ingresar por medio de inicio de sesión, ya sea por una imagen o por datos, como alguna contraseña o número de cédula, haciendo que este al utilizar sea de una forma más sencilla y ordenada.

Lo que se requiere son listas globales y diccionarios privados para poder guardar la informacion de los usuarios, creamos funciones para registrar a los usuarios y que sean llamados donde se necesitan, esto para evitar crear un código extenso, establecimos un menú de manera que sea mas fácil para el usuario poder registrarse, iniciar sesión o salir del sistema.

Requerimientos:

1. El sistema permitirá realizar un registro, dependiendo si es médico o

paciente.

2. El sistema permitira iniciar sesion dependiendo si es médico o paciente.
3. El sistema permitirá reconocer personas registradas en una base de datos mediante una imagen.
4. El sistema permitirá consultar el face ID y las emociones de una persona mediante el análisis una imagen.

## 2.2. Solución del problema

Para poder crear el registro, creamos dos funciones con le nombre de registropaciente y registromedico, las cuales contienen lo que le vamos a pedir a cada usuario dependiendo si es médico o paciente, para poder guardar la informacion requerida, la guardamos en diccionarios privados y luego en listas globales.

Creamos un ciclo while true para que el usuario tenga tres opciones, registrarse, iniciar sesión o salir, si el usuario ingresa con uno, quiere registrarse, aqui se pide que vuelva a escoger si es un médico o un paciente, una vez realizado el registro, los datos se guardan en un diccionario privado y luego en una lista pública, el usuario procede a iniciar sesion, aqui tiene dos opciones, ingresar como médico o ingresar como paciente, nuevamente tiene dos preferencias, si desea ingresar con un usuario y contraseña o con una imagen, si fuera el caso de ser médico o ingresar con el número de cédula o con una imagen si fuera el caso de ser paciente.

Si el usuario desea salir del sistema, tiene la opcion tres, donde se realiza un break para poder romper el ciclo.

## 2.3. Análisis de resultados

Se presenta un correcto ingreso al sistema, es decir, este permite que el usuario pueda iniciar sesión o registrarse, además, permite reconocer a los usuarios por medio de sus datos y por medio de la imagen de su rostros, por último existe la posibilidad de salir del sistema si fuera el caso de que el usuario no quiera estar más en el.

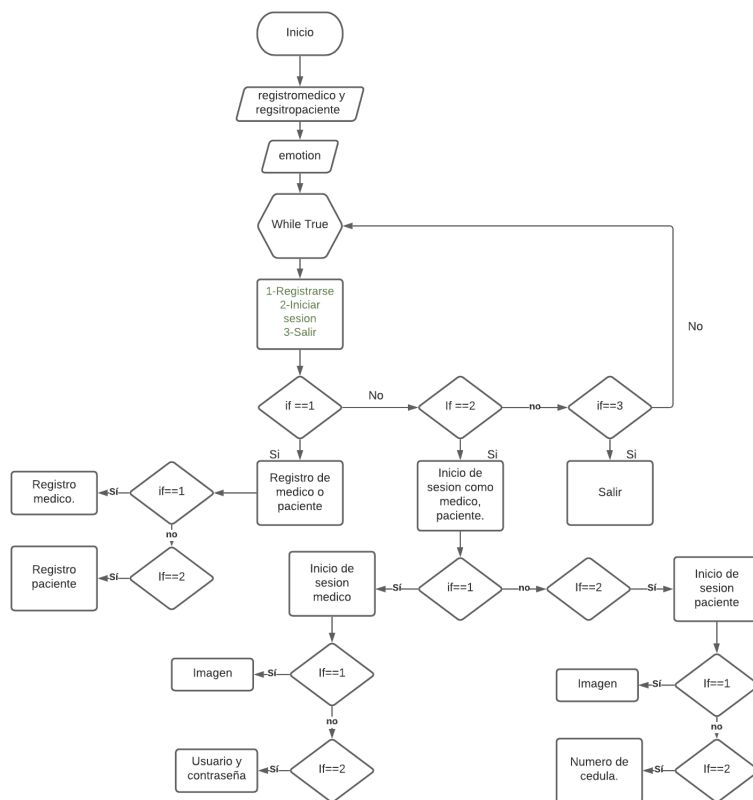


Figura 2.1: Diagrama de flujo de la aplicación (I Etapa).

Cuadro 2.1: Análisis de resultados con respecto a los requerimientos (I Etapa).

Requerimiento	Estado	Observaciones
1. El sistema permitirá realizar un registro, dependiendo si es médico o paciente	Completado (100 %)	
2. El sistema permitira iniciar sesion dependiendo si es médico o paciente	Completado (100 %)	
3. El sistema permitirá reconocer personas registradas en una base de datos mediante una imagen	Completado (100 %)	
4. El sistema permitirá consultar el face ID y las emociones de una persona mediante el análisis una imagen	Completado (100 %)	

## 2.4. Discusión

Durante la realización del algoritmo del sistema, se presentaron algunos problemas, uno de ellos fue el de no poder almacenar los datos en listas o diccionarios, no se comprendía como debía realizarse si por medio de los dos o solo con uno, el que más se presento fue el algoritmo del reconocimiento facial, no se podía sacar el código de rostro de la función ya que era privada, además, a la hora de realizar el ingreso con la imagen este no lograba leer el código rostro de esta, dando errores de que no reconocia a la persona. Estos se resolvieron por medio de consultas al profesor y con la ayuda de una estudiante de la misma carrera que esta en penúltimo semestre.

# Capítulo 3

## II Etapa

### 3.1. Identificación y análisis del problema

En la segunda etapa se requiere archivos para poder guardar la información de manera permanente ya que antes solo se guardaba cierto tiempo. Además el usuario podrá obtener, editar o eliminar los datos guardados anteriormente y crear citas o revisar citas.

Para elaborar lo mencionado anteriormente, se crearon funciones y archivos txt en Python que guardaran cada uno de los datos que se le indicaban, para que al invocar las funciones dentro del algoritmo en el menú, el programa a la hora de ser utilizado pueda solicitar los datos necesarios al usuario y guardar esos datos de manera permanente y solicitarlos al momento que desee el usuario, también, se realizó una función para que al paciente se le pueda hacer un diagnóstico de sus enfermedades.

1. El sistema permitirá un registro de un usuario, ya sea médico o paciente.
2. El sistema permitirá un inicio de sesión de un usuario, ya sea médico o paciente.
3. El sistema permitirá guardar información del usuario en una base de datos.
4. El sistema permitirá que los usuarios consulten, editen o eliminen información guardada anteriormente.
5. El sistema permitirá que los usuarios realicen citas, donde sean analizadas por una imagen.

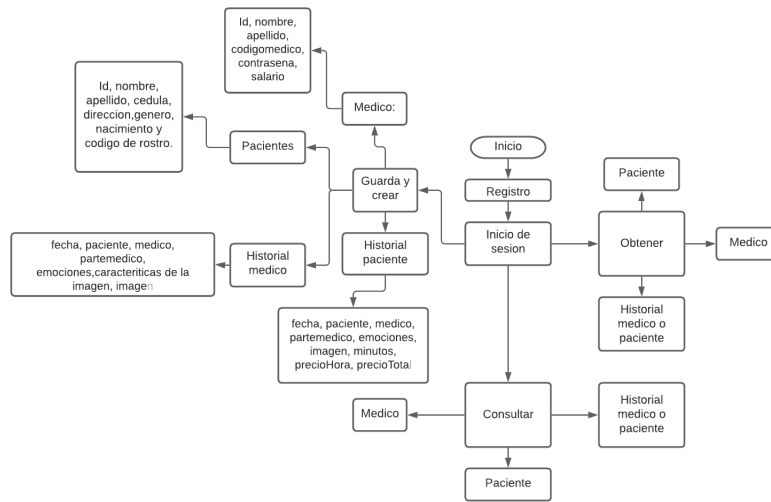


Figura 3.1: Diagrama de flujo de la aplicación (II Etapa).

### 3.2. Solución del problema

El usuario podrá guardar información de manera permanente en archivos, además, podrá consultar, editar o eliminar los datos guardados anteriormente. De la misma manera, si el usuario es paciente, podrá crear citas, donde se le indique por medio de una foto las emociones, características y enfermedades, estos van a ser guardados en los archivos, para que el paciente pueda acceder a su diagnóstico de enfermedades, así, como a sus citas.

Si el usuario es médico, este podrá obtener, editar y eliminar datos guardados anteriormente en archivos, como es un médico, este podrá tener acceso al historial del paciente y a toda la información o datos personales del paciente.

Todo esto se realizó, por medio de funciones de crear, donde se invocaba a la carpeta realizada llamada Datos que contenían archivos txt, para que se guardaran los datos en esos archivos, en la función agregar se le solicitaba a los usuarios los datos necesarios como su nombre, apellido, número de cédula, el código rostro, dirección, género, una contraseña, entre otros, en la función obtener se solicitaban los datos de los usuarios agregados para que después estos puedan ser vistos, en la función editar permite al usuario corregir su información personal, en existir permite analizar si el usuario ya está agregado a la base de datos, y por último la función eliminar permite borrar de la base de datos a algún usuario.

### 3.3. Análisis de resultados

Se realizó de manera ordenada archivos con sus respectivas funciones de crear, agregar, obtener, editar, existir, eliminar. De la misma forma con los historiales de medico y paciente, permitiendo que la información se guarde en archivos de manera permanente. Además, se podrán crear citas con su respectivo diagnostico realizados mediante las emociones obtenidas de la imagen por medio del análisis del reconocimiento facial.

Cuadro 3.1: Análisis de resultados con respecto a los requerimientos (II Etapa).

Requerimiento	Estado	Observaciones
1. El sistema permitirá un registro de un usuario, ya sea medico o paciente.	Completo.	
2. El sistema permitira un inicio de sesion de un usuario, ya sea medico o paciente.	Completo.	
3. El sistema permitira guardar informacion del usuario en un base de datos.	Completo.	
4. El sistema permitira que los usuarios consulten, editen o eliminen informacion guardada anteriormente.	Completo.	
5. El sistema permitira que los usuarios realicen citas, donde sean analizadas por una imagen.	Completo	

### 3.4. Discusión

En la segunda etapa se presentaron problemas con los archivos, no se lograba recorrer de manera correcta los txt, el segundo problema que se tuvo, fue con el inicio de sesión con imagen no se lograba ingresar porque

cambiaba el Id, pero se realizo con el top, el tercer problema fue con la función de editar, ya que no se modificaba porque no se estaban comparando los datos ingresados con los del archivo. Resolvimos la mayor parte de los problemas por medio de debugger y con la ayuda del profesor del curso por medio de consultas fuera de clases.



# Capítulo 4

## III Etapa

### 4.1. Identificación y análisis del problema

En la tercera etapa se requiere elaborar una plataforma que guarde todos los datos de los usuarios, así como mostrarlos, editarlos e eliminarlos, se requiere crear una factura, obtenerla y cambiar el estado, además, se debe obtener las estadísticas de los pacientes atendidos, enfermedades, enfermedades con género, emociones más presentes, personas enfermas y sanas.

Esto se realizara con Python, por medio de interfaz gráfica, donde se utilizaran ventanas y botones, para poder llevar a cabo todo lo solicitado en el problema, también se utilizaran clases para la facturación y estadísticas y para desarrollar la interfaz. Para poder guardar los datos se usaron archivos txt, para después acceder a ellos para mostrar los datos a los usuarios, además, el usuario puede interactuar con la interfaz de manera sencilla, ya que permite y muestra un ambiente ordenado y de fácil uso.

1. Implementación de interfaz gráfica.
2. El sistema permitirá que los usuarios inicien sesión o se registren por medio de interfaz gráfica.
3. El sistema permitirá que los usuarios obtengan sus enfermedades al ingresar el path de una imagen por medio de interfaz gráfica.
4. El sistema permitirá crear y obtener facturas.
5. Los usuarios podrán crear citas y obtener por medio de interfaz gráfica.
6. El médico podrá obtener estadísticas por medio de interfaz gráfica.

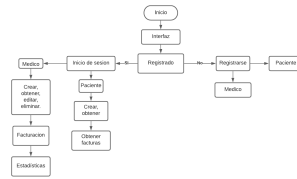


Figura 4.1: Diagrama de flujo de la aplicación (III Etapa).

## 4.2. Solución del problema

Para esta tercera etapa, se inició con la realización de la ventana principal, donde permitiría al usuario escoger como podría iniciar sesión, si como médico o paciente, según la escogencia del usuario se levantará una ventana donde el usuario pueda ingresar su código médico y contraseña si fuera el caso que ingresa como médico, pero si se ingresa como paciente este le solicitará al usuario la cédula y la contraseña para poder ingresar sesión.

Al iniciar sesión, si fuera el caso de que el usuario no se encuentra registrado, este podrá realizarlo, para que después pueda ingresar, cuando este permite el acceso se levantará una ventana diferente para los usuarios.

En el caso del médico, este tiene en el menú las opciones de crear un médico eliminarlo y mostrarlo, crear y obtener historial médico, obtener el historial paciente, crear, obtener y cambiar estatus de la factura y finalmente podrá obtener estadísticas del hospital, las cuales son: pacientes atendidos, enfermedades atendidas, enfermedades por género, desglose de los sentimientos más presentes y por último, obtener cuantas personas son sanas y enfermas.

En el caso del paciente, tiene las opciones de mostrar, editar y eliminar a un paciente, así como crear una cita y realizar un diagnóstico de enfermedades y por último tiene la opción de obtener la factura.

## 4.3. Análisis de resultados

Se realizó una interfaz gráfica donde los usuarios pueden iniciar sesión y registrarse dependiendo si es médico o paciente, los usuarios registrados pueden, crear, obtener, editar, eliminar, crear y obtener facturas, obtener emociones y enfermedades por medio de un reconocimiento facial y por último, los usuarios médicos pueden obtener estadísticas del hospital.

Cuadro 4.1: Análisis de resultados con respecto a los requerimientos (III Etapa).

Requerimiento	Estado	Observaciones
1. Implementación de interfaz gráfica.	Completo.	
2. El sistema permitirá que los usuarios inicien sesión o se registren por medio de interfaz gráfica.	Completo.	
3. El sistema permitirá que los usuarios obtengan sus enfermedades al ingresar el path de una imagen por medio de interfaz gráfica.	Completo	
4. El sistema permitirá crear y obtener facturas.	Completo (90 %)	Solo obtiene una factura y no cambia el estado.
5. Los usuarios podrán crear citas y obtener por medio de interfaz gráfica.	Completo	
6. El médico podrá obtener estadísticas por medio de interfaz gráfica.	Completo (95 %)	Falta obtener desglose de sentimientos.

## 4.4. Discusión

En la tercera etapa, se presentaron muchos problemas, el primer problema que se presentó, fue con el inicio de sesión, porque los datos ingresados en la barra de StringVar no conciden con los registrados en los archivos txt, además se tuvo problema con lo del reconocimiento facial al obtener emociones y para el diagnóstico, ya que al implementarlo en el código alguna ventana dejaba de levantarse o funcionar y los índices cambiaban de rango, otro problema fue el de obtener los datos, ya que no estaba mostrándolos. Estos problemas fueron resueltos con ayuda del profesor y algunos compañeros del curso. Sin embargo, hay problemas que fue imposible solucionarlos, como lo es el obtener factura, solo muestra una factura y cuando se presiona el botón siguiente o anterior no cambia, además, no se logró implementar lo de cambiar el estado de la factura, porque al recorrer los datos del estado en el txt para editarlo, esto no

funcionaba por problemas del recorrido en el mismo al igual que en obtener factura, el ultimo problema que no se logro solucionar fue el de obtener el desglose de los sentimientos más presentes, porque no los muestra y no se logro encontrar el motivo.

# Capítulo 5

## Conclusiones

1. Se logro poner en practica los conocimientos obtenenidos en clase, programación orientada a objetos, clases, funciones, listas, diccionarios, ciclos, condicionales.

2. Se implemento el uso de archivos, por medio de la creación de funciones que ayudaron a que los datos fueran guardados, poder editarlos, eliminarlos y obtenerlos, además, de se logró utilizar el reconocimiento facial para obtener enfermedades, todo esto gracias a a lo visto y aprendido en clases.

3. Implementacion de interfaz grafica, por medio de tkinter, con base a esto, se logró crear ventanas para crear, obtener, editar, eliminar informacion por medios de los Stringvar.

# Referencias

- Moreano, J. A. C., Pulloquina, R. H. M., Lagla, G. A. F., Chisag, J. C. C., Pico, O. A. G. (2017, 06). Reconocimiento facial con base en imagenes. *Bolet  in Redipe*, 6, 143-151. Descargado de <https://revista.redipe.org/index.php/1/article/view/267>
- Poveda, M., Merch  n, F. (2015, 05). Implementaci  n de un sistema de control de acceso basado en reonomiento facial. *Revista Prisma Tecnol  gico*, 6, 34-39. doi: <https://revistas.utp.ac.pa/index.php/prisma/article/view/609>